LDSP: Shopping with Cryptocurrency Privately and Quickly under Leadership

Lucien K. L. Ng, Sherman S. M. Chow,

Donald P. H. Wong, and Anna P. Y. Woo

Department of Information Engineering
Chinese University of Hong Kong (CUHK), Hong Kong

Shopping with Cryptocurrency?

☹️ Slow

- Bitcoins takes 10 mins to include a payment transaction
- (~60 mins to confirm)

☹️ Low Privacy

- All transactions are exposed on the blockchain

Traditional Layer-2 Networks

- 😁 Low-latency Payment
  - Payer & payee confirm their payment "locally" (off-chain)
    - jointly-sign a balance sheet of their (updating) asset & sync "on-chain" later
- ☹️ High collateral: money locked (mostly) for a *single* payee
- ☹️ Constantly-Online Requirement
  - Payers & payees need to monitor the on-chain transactions
    - worry that the other party might upload an "outdated" balance sheet
- ☹️ (Still) Low Privacy
  - The final balance sheet is exposed on the blockchain
    - total transaction amount & who paid to whom are leaked

LIGHTNING NETWORK

Snappy: Layer-2 Solution for Retail

- *Unidirectional* system tailored for *retail* payments [NDSS 20]
  - Users take the role of either customer/payer or merchant/payee
- 😁 Low-latency Payment Solution (~ layer-2)
- 😁 Offline Customer (no need to monitor on-chain)
  - Merchant has disincentive in uploading outdated sheet
    - Penalty from their collaterals
- 😁 Low Collaterals
  - Customer collateral is "shared" among all merchants in Snappy

Shortcomings of Snappy

☹️ No Privacy (Merchants share all Payer Spending Histories)

- State = all payer's spending histories
- Merchants also serve as statekeepers
- Confirmation needs 51% of merchants to vouch
  - Each checks the balance of the payer
  - pays back from their collateral if vouched wrongly
    - e.g., a double-spending transaction

LDSP: Layer-2 Anonymous Payment

😁 **L**ow Latency

- Speedy confirmation of off-chain payment within *seconds*

😁 **D**ynamic and Distributed Setup

- Merchants (payees) can join and leave dynamically
- Customers (payers) can pay multiple merchants

😁 **S**calable On-chain Processes

- Our design naturally supports batching to *reduce on-chain costs*

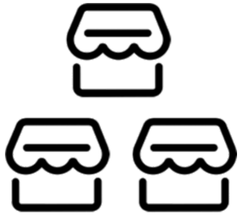
😁 **P**rivacy

- Customers (payers) can hide their identities
- They can "hide in the crowd" to obfuscate the payment amount

Core Entities in LDSP

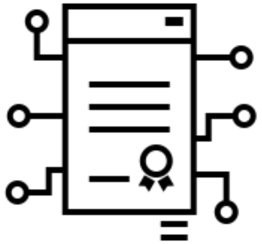
- Customers, who want to pay merchants (off-chain)

- A consortium, formed by a group of merchants



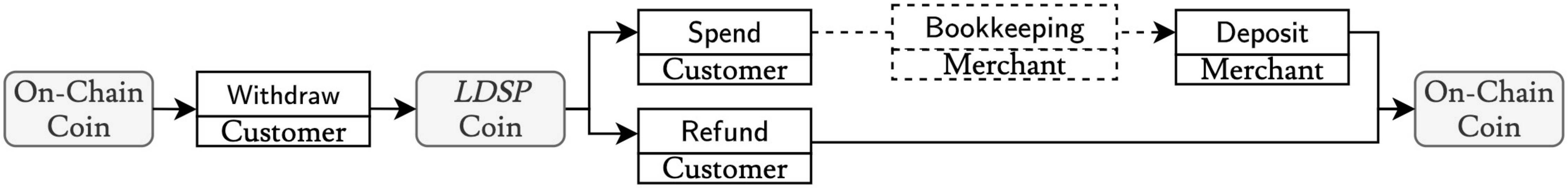
- A *leader*, leading the consortium

- An arbiter (smart contract), for resolving disputes, *etc*.



Workflow (& Core Functions)

1. Customers *withdraw* LDSP coins by funding them on-chain
2. Customers *spend* (off-chain) LDSP coins to a merchant
3. Customers *refund* unspent LDSP coins
4. Merchants *bookkeep* and *deposit* the received LDSP coins to receive the on-chain coins (funded by customers)



Dilemma: Who issue LDSP coins?

- All merchants (or statekeepers) are needed to issue coins
  - Not scalable!
- Only 1 merchant (or statekeeper) is needed to issue coins
  - Keep issuing coins to a "customer" to be spent at victim merchant
  - Becomes a money-printing factory!
- A large subset of them needed to issue coins
  - The worst of both worlds?
  - Need many to help, but they can still collude
  - Or is it?

High-Level Operations of LDSP

- We introduce *leaders,* each leads a group of merchants
- Merchants in a group jointly issue coins, forming a "*virtual bank*"
- A coin can either be spent with the issuing merchant
- or at another merchant, which we call cross-group payments

Leader’s Duties and Motivation

- settles w/ other leaders for cross-group payments
- confirms payment (in its group) to avoid double-spending
- motivations: getting service fees, establishing partnerships

Highlights of LDSP

- Small group size, which mitigates the scalability problem
- State = which merchant gets back how many LDSP coins
  - Formed by "consensus" between the leaders and the merchants
- Merchants can belong to different "virtual banks"
  - *i.e.*, the groups forming the banks are *overlapping*
- Merchants, who are payees, have no incentive to forge

Design Intuition (Merchant Perspective)

- Self-evolving ecosystem
- We leverage the "business relationship" among merchants to help their beloved customers.
- Somewhat like how inter-bank transactions are cleared
  - *"I know this virtual bank well. I'll just accept its coins."*
  - *"I'm not familiar with this bank. Let me talk to them first before accepting too many coins issued by them."*

Design Intuition (Customer Perspective)

- Somewhat similar to customer-loyalty programs
- Think of the coins are the "points"/"mileages"

- *"I frequent these shops. Let me get more coins from them."*
- *"I may occasionally buy this each month. I'll get less here."*

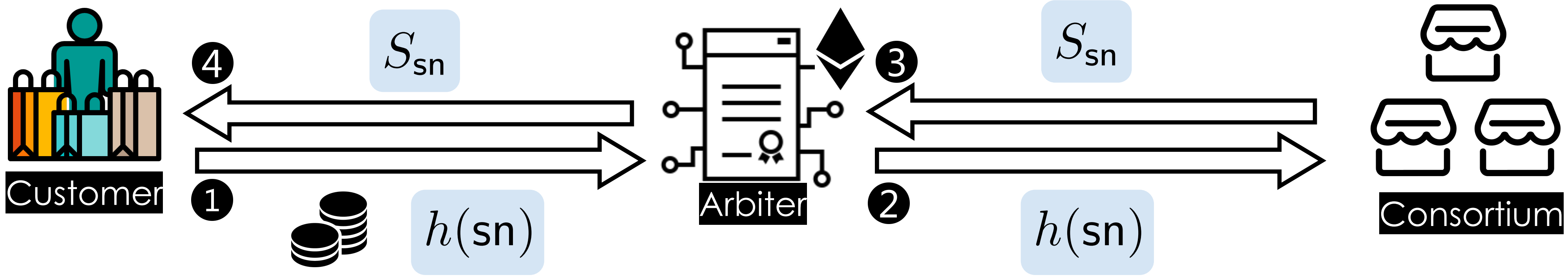
- LDSP coins are partially blind signatures from a group
- "80/20 rule": trade a bit of privacy for better efficiency

A bit more details

Withdrawal, Payment, Refund, ...

Customer's Withdrawal

- The customer funds LDSP coins via the arbiter
- The customer uploads a blinded h(sn) w/ an on-chain coin
  - h: crypto hash function, sn: (random) serial number
- The merchants *jointly* sign, *blindfolded*, on the h(sn) as S_{sn}
- The customer gets the signature S_{sn} as an LDSP coin

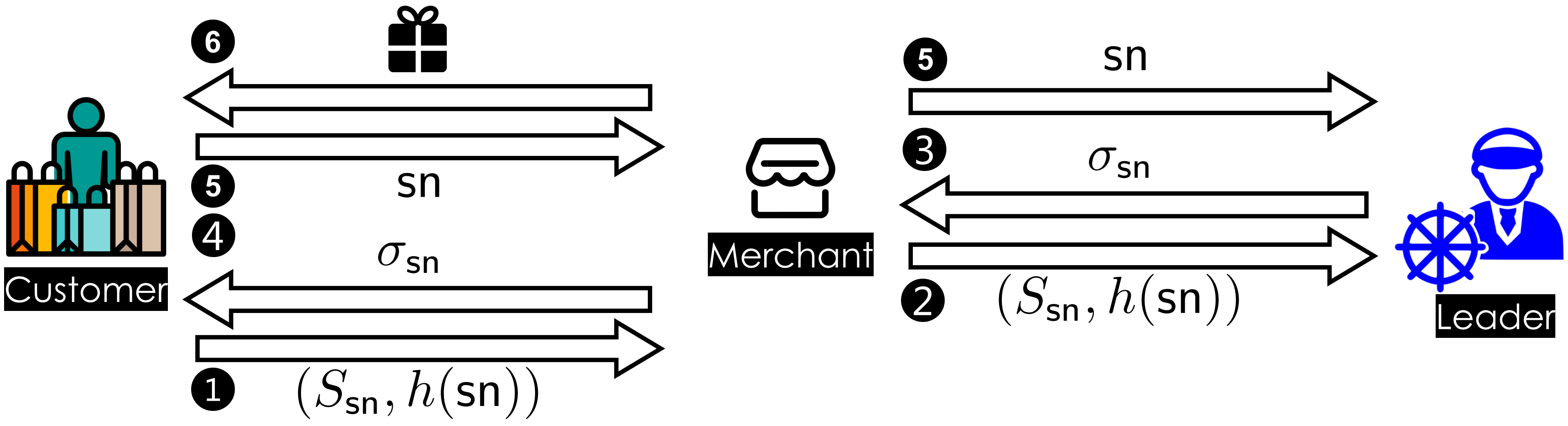


Payer Privacy

- Basically, we adopt a multi-blind signature approach
- Hide the link btw. on-chain coins & LDSP (off-chain) coins
- The customers thus hide among those spending coins from the same virtual bank

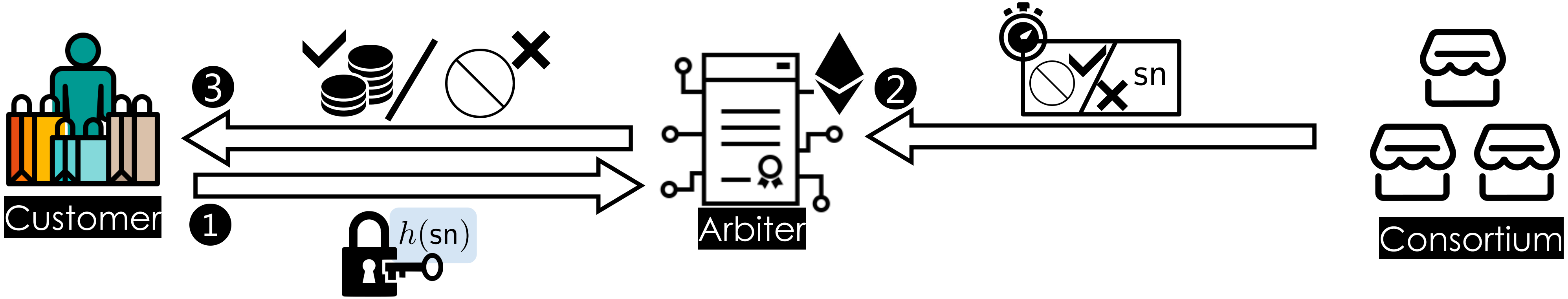
Customer's off-chain Payment

- Customer reveals the signed h(sn) to the merchant & leader
- The leader confirms it with its signature σ_{sn}
- The customer reveals sn to the merchant



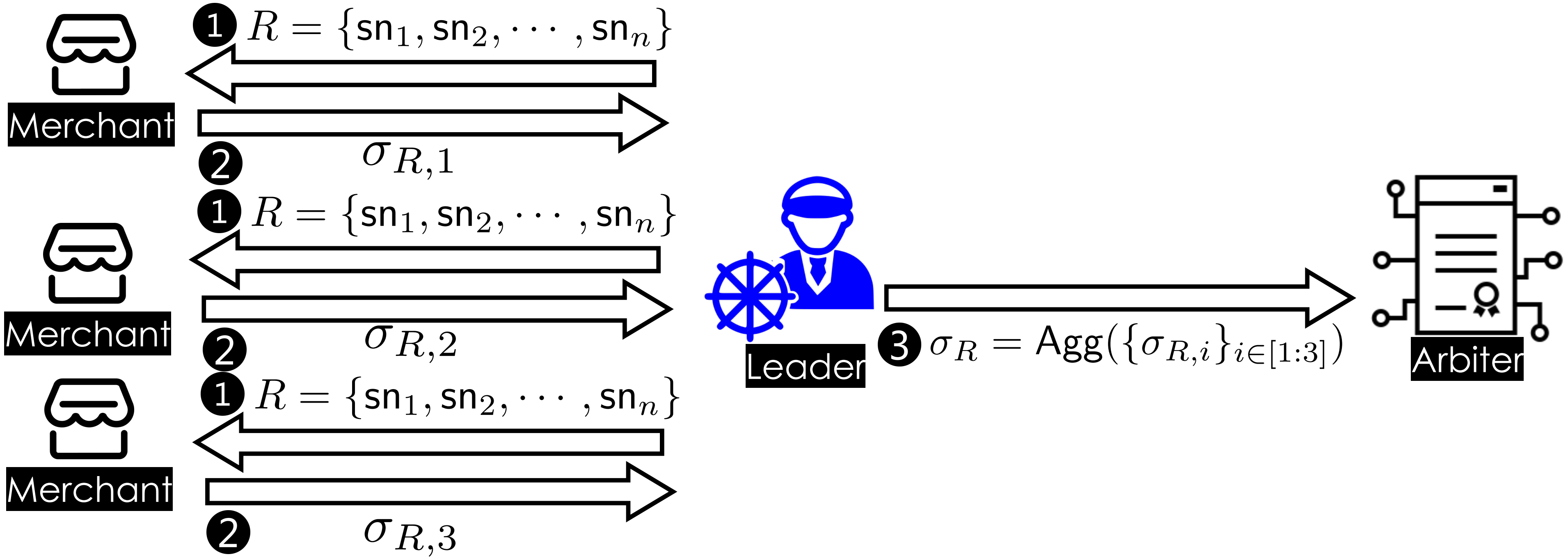
Customer's Refund

- The customer uploads the opening for $h(\text{sn})$ to the arbiter
- If sn has been spent, the merchant can reveal sn to debunk
- If no dispute, the customer gets back the on-chain coin



Merchants' Bookkeeping (& Why no Double-Spending)

- The leader releases all payment records in batch
  - which is uploaded to the arbiter after the merchants *jointly* signing on it
- The leader will be blamed if any double-spent coin is spotted
- The merchants can deposit all bookkept coins altogether later

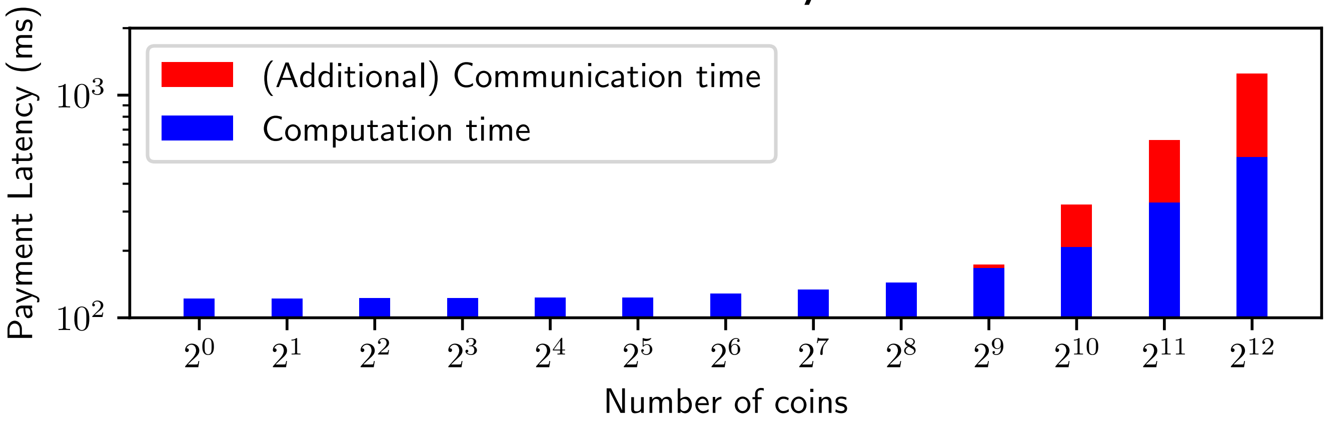


LDSP's Safety (No one loses money)

- Safety of the Customers:
  - They can always refund an unspent coin
- Safety of the Merchants:
  - For double-spending, they can blame the leader for compensation
  - No merchant can mint coin w/o the signature of all other merchants

Low-Latency Payment & Low On-Chain Cost

- Off-chain payment with < 512 coins is done in < 0.5s
  - Urban Network: 100 Mbps Bandwidth, ~20ms Latency



- Low on-chain cost

Operation	Gas (on Ethereum)	USD
Baseline	21000/tx	3.72/tx
(Batched) Withdrawal	38.36/coin	0.0068/coin
(Batched) Refund	25.76/coin	0.0046/coin

based on the average gas price and exchange rate on 1 May, 2021

Summary

- LDSP: Shopping with Cryptocurrency *Privately* and *Quickly* under "Consortium Leadership"
  - a Low Latency, Dynamic & Distributed System
  - w/ Scalable On-chain Process, & Payer Privacy
- What's more in the paper:
  - The use of round and epoch
  - Keeping a low on-chain cost via batching
  - Analysis on LDSP's safety & liveness
  - How LDSP achieves low collaterals, avoids single-point-of-failure, *etc*.
- Contact: `{luciengkl, sherman}@ie.cuhk.edu.hk`